LOCAL ERASURE MAP DECODER

Field of the Invention

The present invention relates to a method for decoding at least one codeword, the at least one codeword having been generated by an encoder comprising a structure providing a code representable by a set of branch transitions in a trellis diagram. Further, the present invention provides a respective decoder, as well as a mobile station and a base station in a communication network employing the decoder. Moreover a communication system comprising the base stations and mobile stations is provided.

Technical Background

Shift-Register Coding

Convolutional codes and related codes may be generated by means of one or more cascaded or concatenated shift registers. For matters of simplicity, binary shift registers are considered in the following sections. The binary shift registers are capable of taking the value of either binary 0 or binary 1. When a shift occurs, the content of each register is forwarded to the subsequent register to be its new content. Usually the input to the encoder is used as the new content of the first register.

The output of a binary shift register encoder is usually obtained by modulo-2 additions of several shift register contents prior to shifting. As an illustration, a simple binary shift-register encoder is shown in Fig. 1, where the number of shift registers $r = 2$ and the number of states is $M = 4$. Each shift register is represented by a D, and each modulo-2 addition unit is represented by "+". Two output bits are obtained from one input bit: The first output bit is identical to the input bit (upper branch), while the second output bit is obtained by modulo-2 addition of the shift register states and the input bit (lower branch).

In Fig. 2 a state-transition diagram for the encoder from Fig. 1 is shown. Each state is represented by the values of the shift register. Each transition is represented by a directed edge. A transition caused by an input bit of zero is denoted by a broken edge, while a transition caused by an input bit of one is denoted by a straight edge. Each edge is further labeled with the input bit followed by the corresponding output bits. An alternative representation of the state transition diagram is a trellis, which is composed of trellis elements as shown in Fig. 3. Further details about shift-register encoding (also known as convolutional encoding) may for example be found in Lin et al., "Error Control Coding: Fundamentals and Applications", Prentice-Hall Inc., chapter 10.

Shift registers are commonly employed for convolutional codes. Recently, they have also been used in "turbo codes" reaching very low error rates, which make them attractive for communication systems.

Popular decoding algorithms for shift-register codes are for example the Viterbi algorithm and the maximum a-posteriori algorithm. While the former is often used for traditional convolutional codes, the latter is very popular for the decoding of turbo codes due to its soft a-posteriori probability output.

Maximum A-Posteriori Algorithm

A brief description of the maximum a-posteriori algorithm is provided in the following paragraphs. For brevity, the binary case is considered in more detail. The extension to the non-binary case should impose no problem to those skilled in the art. Generally speaking in the non-binary case event probabilities may usually not be expressed by a log-likelihood ratio. Instead some (possibly logarithmic) absolute probability measure may be used. Evidently, all equations given subsequently involving log-likelihood ratios would have to be changed so that they hold for mentioned absolute probability measures.

A simplifying characteristic of the binary case is that - since there are only two possible events - the event probabilities may be expressed in terms of a log-likelihood ratio (LLR), which is generally defined by

$$LLR = \ln\frac{p(x=1)}{p(x=0)} = \ln\frac{p(x=1)}{1-p(x=1)} \qquad \text{Equation 1}$$

as the natural logarithm of the ratio of probabilities that x is one of the two possible events.

The following symbols are used throughout this document:

k	Information bit index
K	Number of information bits in one coded block

r	Number of shift registers in the encoder
M	Number of states within the encoder
S_k	State for index k
d_k	information bit number k, either 0 or 1, prior to encoding
$\hat{d}_k$	information bit number k, either 0 or 1, after decoding
x_k^s	Systematic value for bit d_k at the output of encoder, -1 or +1
x_k^p	Parity value for bit d_k at the output of encoder, -1 or +1
$x_k = (x_k^s x_k^p)$	Systematic and parity value sequence for bit d_k at the output of encoder
y_k^s	Received value for systematic bit k at the input of decoder
y_k^p	Received value for parity bit k at the input of decoder
$y_k = (y_k^s y_k^p)$	Received systematic and parity bit sequence for information bit k at the input of decoder
$\gamma_{k,i}(y_k, m', m'')$	Branch transition probability for transit between states m' and m'', given the observation of the received codeword y_k, assuming an information bit d_k =i (see explanation of Equation 2)
$\Gamma_k(y_k, m', m'')$	Logarithm of $\gamma_{k,i}$
$\alpha_k(S_k)$	Probability measure for being in state S_k for information bit k given the received sequence $y_1 \ldots y_k$
$\beta_k(S_k)$	Probability measure for being in state S_k for information bit k, given the received sequence $y_k \ldots y_K$
$L^i(x_k^s)$	Intrinsic (a-priori) probability log-likelihood ratio which is available for bit x_k^s
$L^e(x_k^s)$	Extrinsic probability log-likelihood ratio which is computed for bit x_k^s
$L(x_k^s)$	Decision (a-posteriori) probability log-likelihood ratio which is computed for bit x_k^s

The algorithm has two components commonly referred to as the forward and backward recursion. More specifically, two distributions, α_k and β_k are recursively updated. The quantity $\alpha_k(S_k)$ represents the probability measure for being in state $S_k = m$ for information bit k, given the received sequence $y_1 \ldots y_k$. In a similar manner, $\beta_k(S_k)$ represents the probability measure for being in state $S_k = m$ for information bit k, given the received sequence $y_k \ldots y_K$.

Both recursions may be defined based on the so-called branch transition probability $\gamma_{k,i}(y_k, m', m'')$. This represents the probability to transit between states m' and m'' given the observation of the received codeword y_k, assuming that the information bit causing the transit is d_k =i. The branch transition probability can be computed as

$$\gamma_{k,i}\left(\left(y_k^s, y_k^p\right), S_{k-1}, S_k\right) = q\left(d_k = i \middle| S_{k-1}, S_k\right) \cdot p\left(y_k^s \middle| d_k = i\right) \cdot p\left(y_k^p \middle| d_k = i, S_{k-1}, S_k\right) \cdot \Pr\left\{S_k \middle| S_{k-1}\right\}$$

Equation 2

The value of $q\left(d_k = i \middle| S_{k-1}, S_k\right)$ is either one or zero, depending on whether bit i is associated with the transition from state S_{k-1} to state S_k or not. $\Pr\left\{S_k \middle| S_{k-1}\right\}$ is the a priori probability of the information bit d_k. In the context of turbo decoding this probability may be the obtained extrinsic information from another decoder. Other terms can be derived easily by those skilled in the art. For example if no a priori information is available the probabilities may be set equal.

Equation 2 can be simplified by omitting the index i if it is assumed that the γ values exist only for those transitions where $q\left(d_k = i \middle| S_{k-1}, S_k\right) = 1$. Using this assumption the equation can be rewritten as

$$\gamma_k\left(\left(y_k^s, y_k^p\right), S_{k-1}, S_k\right) = p\left(y_k^s \middle| x_k^s\right) \cdot p\left(y_k^p \middle| x_k^s, S_{k-1}, S_k\right) \cdot \Pr\left\{S_k \middle| S_{k-1}\right\}$$

Equation 3

Considering the case in which for each information bit d_k at the encoder input, there are two coded bits $x_k = (x_k^s x_k^p)$ at the output of the encoder, Equation 3 may be further simplified, resulting in a code rate of $\frac{1}{2}$. Furthermore, when considering the binary case, Equation 3 may be further simplified by using logarithmic expressions:

$$\Gamma_k\left(\left(y_k^s, y_k^p\right), S_{k-1}, S_k\right) = \ln \gamma_k\left(\left(y_k^s, y_k^p\right), S_{k-1}, S_k\right) \qquad \text{Equation 4}$$

In case of a binary shift-register code, the number of states M can be computed as

$$M = 2^r \qquad \text{Equation 5}$$

Initialization

For each branch transition originating in state S_{k-1} ending in state S_k the branch transition probability for a BPSK (Binary Phase Shift Keying) AWGN (Additive White Gaussian Noise) case is given by

$$\Gamma_k\left(\left(y_k^s, y_k^p\right), S_{k-1}, S_k\right) = \frac{1}{2} \cdot x_k^s \cdot \left(L^i\left(x_k^s\right) + L_c y_k^s\right) + \frac{1}{2} L_c y_k^p x_k^p \qquad \text{Equation 6}$$

with k running from 1 to K.

Since the last term is used frequently below, Equation 6 may be rewritten as

$$\Gamma_k\left(\left(y_k^s, y_k^p\right), S_{k-1}, S_k\right) = \frac{1}{2} \cdot x_k^s \cdot \left(L^i\left(x_k^s\right) + L_c y_k^s\right) + \Gamma_k^e\left(y_k^p, S_{k-1}, S_k\right) \qquad \text{Equation 7}$$

using

$$\Gamma_k^e\left(y_k^p, S_{k-1}, S_k\right) = \frac{1}{2} L_c y_k^p x_k^p \qquad \text{Equation 8}$$

L_c is a channel scaling factor which may be derived from the signal-to-noise ratio (SNR), and is in this case

$$L_c = \frac{2}{\sigma^2} \qquad \text{Equation 9}$$

with σ^2 representing the channel noise variance.

The initial values of α_k and β_k may be initialized according to system parameters. For a code which begins and ends in the state $m = 0$, the initializations should be

$$\alpha_0\left(S_0\right) = \begin{cases} 0 & \text{for } \mathrm{S}_0 = 0 \\ -\infty & \text{else} \end{cases} \qquad \text{Equation 10}$$

and

$$\beta_K(S_0)=\begin{cases}0 & \text{for } S_0=0\\ -\infty & \text{else}\end{cases} \qquad \text{Equation 11}$$

Forward Recursion

For each state S_k, k running from 1 to K, α_k may be calculated as

$$\alpha_k(S_k)=\ln\frac{\sum\limits_{S_{k-1}=1}^{M}\exp(\alpha_{k-1}(S_{k-1})+\Gamma_k(y_k,S_{k-1},S_k))}{\sum\limits_{S_k=1}^{M}\sum\limits_{S_{k-1}=1}^{M}\exp(\alpha_{k-1}(S_{k-1})+\Gamma_k(y_k,S_{k-1},S_k))} \qquad \text{Equation 12}$$

Backward Recursion

For each state S_k, k running from $K-1$ to 0, β_k may be calculated as

$$\beta_k(S_k)=\ln\frac{\sum\limits_{S_{k+1}=1}^{M}\exp(\beta_{k+1}(S_{k+1})+\Gamma_{k+1}(y_{k+1},S_k,S_{k+1}))}{\sum\limits_{S_k=1}^{M}\sum\limits_{S_{k+1}=1}^{M}\exp(\beta_{k+1}(S_{k+1})+\Gamma_{k+1}(y_{k+1},S_k,S_{k+1}))} \qquad \text{Equation 13}$$

Decoding

A full decoding process may consist of an application of the forward and backward recursion. After these recursions one can update the soft-output decision (i.e. the posteriori probability) of each information bit:

$$L^e(x_k^s)=\ln\frac{\sum\limits_{(m',m'')\in S^+}\exp(\alpha_{k-1}(m')+\Gamma_k^e(y_k,m',m'')+\beta_k(m''))}{\sum\limits_{(m',m'')\in S^-}\exp(\alpha_{k-1}(m')+\Gamma_k^e(y_k,m',m'')+\beta_k(m''))} \qquad \text{Equation 14}$$

$$L(d_k)=L_c\cdot y_k^s+L^i(x_k^s)+L^e(x_k^s) \qquad \text{Equation 15}$$

In the above equation, S^+ is the set of ordered pairs (m',m'') corresponding to all state transitions $m'\longrightarrow m''$ which are caused by data input $d_k=1$. S^- is similarly defined for $d_k=0$.

$$S^{+} = \left\{ (m', m'') \middle| m' \xrightarrow{d_k = 1} m'' \right\}$$

Equation 16

$$S^{-} = \left\{ (m', m'') \middle| m' \xrightarrow{d_k = 0} m'' \right\}$$

Equation 17

Using Equation 15 the value of the k^{th} transmitted bit can be estimated as

$$\hat{d}_k = \begin{cases} 1 & \text{if } L(d_k) \geq 0 \\ 0 & \text{if } L(d_k) < 0 \end{cases}$$

Equation 18

It should be noted that the extrinsic quantity L^e obtained in Equation 14 may be used as intrinsic information for a subsequent decoder. Likewise the quantity L^i in Equation 15 may have been obtained as intrinsic information from the extrinsic information of another decoder.

Those skilled in the art will recognize that both quantities can also be set to proper values in case no information is available from another decoder. Further details about applicability of the algorithm to turbo codes, intrinsic information and extrinsic information are given in Berrou et al., „Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes (1)“, Proc. IEEE Int. Conf. On Communications, pp. 1064-1070, May 1993.

Max-Log-MAP Algorithm

To simplify the calculations involved, Equations 12 and 13 may be approximated and substituted by

$$\alpha_k(S_k) = \frac{\max\limits_{S_{k-1}=1..M} \left(\alpha_{k-1}(S_{k-1}) + \Gamma_k(y_k, S_{k-1}, S_k) \right)}{\max\limits_{S_k=1..M} \left[\max\limits_{S_{k-1}=1..M} \left(\alpha_{k-1}(S_{k-1}) + \Gamma_k(y_k, S_{k-1}, S_k) \right) \right]}$$

Equation 19

and

$$\beta_k(S_k) = \frac{\max\limits_{S_{k+1}=1..M} \left(\beta_{k+1}(S_{k+1}) + \Gamma_{k+1}(y_{k+1}, S_k, S_{k+1}) \right)}{\max\limits_{S_k=1..M} \left[\max\limits_{S_{k+1}=1..M} \left(\beta_{k+1}(S_{k+1}) + \Gamma_{k+1}(y_{k+1}, S_k, S_{k+1}) \right) \right]}$$

Equation 20

Likewise the decision variable can be obtained by modifying Equation 14 to

$$L^{e}\left(x_{k}^{s}\right)=\max_{(m',m'')\in S^{+}}\left(\alpha_{k-1}(m')+\Gamma_{k}^{e}(y_{k},m',m'')+\beta_{k}(m'')\right) \\ -\max_{(m',m'')\in S^{-}}\left(\alpha_{k-1}(m')+\Gamma_{k}^{e}(y_{k},m',m'')+\beta_{k}(m'')\right) \qquad \text{Equation 21}$$

These approximations may degrade the performance of the decoding however.

As can be seen from the equations for the forward and backward recursion, the information from numerous values is involved which is ultimately derived from the received vector corresponding to the transmitted codeword. In a noisy channel environment, chances are high that several received values carry wrong information, which implies that wrong information can be inferred from these values and propagate through the decoding iterations.

Summary of the Invention

It is therefore the object of the present invention is to reduce the influence of such wrong information.

The object is solved by the subject matters of the independent claims. Advantageous embodiments of the present invention are subject matters to the dependent claims.

According to one aspect of the present invention not all information in the forward and/or backward recursion is processed, as it would be required by the respective prior-art equations. According to this embodiment of the present invention some of the terms are excluded instead. The decision, which term/s is/are excluded may for example be determined according to its/their reliability. I.e. a term which would produce degrade the decoding performance when employed in determining the forward and/or backward recursion is omitted from the respective equation.

In one of the different exemplary embodiments of the present invention, a method for decoding at least one codeword, wherein the at least one codeword has been generated by an encoder comprising a structure providing a code representable by a set of branch transitions in a trellis diagram is provided.

According to this embodiment, the method may comprise the steps of initializing a set of branch transition probabilities in the decoder based on the received codeword and the encoder structure, initializing a first probability distribution and a second probability distribution according to the initial state of the encoder used to encode the at least one codeword, recalculating the values of the first probability distribution based on the initial values of the first probability distribution and the set of branch transition probabilities using a recursive algorithm, recalculating the values of the second probability distribution based on the initial values of the second probability distribution and the set of branch transition probabilities using a recursive algorithm, and reconstructing a decoded codeword based on the received codeword and an extrinsic probability measure calculated based on the set of branch transition probabilities, the first and the second probability distribution.

In either each of or both steps of recalculating the values of first or second probability distribution a subset of initial values of the first probability distribution or the second probability distribution, respectively, and a subset of the set of branch transition probabilities may be used for recalculating the respective probability distribution. Further, only the values in the subsets fulfilling a predetermined reliability criterion are used.

In a further embodiment, the encoder may be representable by a shift register structure containing at least one of feed-forward mathematic operations and feed-back mathematic operations.

Moreover, in another embodiment of the present invention, the code is suitable for decoding by employing a maximum a-posteriori algorithm.

In a further embodiment of the present invention, the method may further comprise the step of using an intrinsic probability measure to initialize the set of branch transition probabilities.

Another embodiment of the present invention encompasses the step of using an intrinsic probability measure to reconstruct the decoded codeword.

In a further variation of this embodiment, a decoder representable by two separate decoder instances is used for decoding the at least one codeword in a first decoding step and the method may further comprise the step of using the extrinsic probability measure of the first decoder instance as the intrinsic probability measure in the second decoder instance.

In another variation of this embodiment the method further comprises the step of performing a second decoding iteration in the first decoder instance, wherein the decoder instance uses the extrinsic probability measure of the second decoder instance as the intrinsic probability measure.

According to a further embodiment of the present invention the reliability criterion may be based on at least one of channel estimations of a radio channel via which the at least one codeword has been received, the absolute values of the elements of the first and/or second probability distribution, the number of decoding steps performed and a random process. In another variation the reliability criterion may not be fulfilled by an element of the first or the second probability distribution, if the signal to noise ratio for the element and/or the absolute value of the element is below a predetermined threshold value.

Moreover, the present invention provides in another embodiment, a decoder for decoding at least one codeword, wherein the at least one codeword has been generated by an encoder comprising a structure providing a code representable by a set of branch transitions in a trellis diagram.

The decoder may comprise processing means for initializing a set of branch transition probabilities in the decoder based on the received codeword and the encoder structure, initializing a first probability distribution and a second probability distribution according to the initial state of the encoder used to encode the at least one codeword, recalculating the values of the first probability distribution based on the initial values of the first probability distribution and the set of branch transition probabilities using a recursive algorithm, recalculating the values of the second probability distribution based on the initial values of the second probability distribution and the set of branch transition probabilities using a recursive algorithm, and for reconstructing a decoded codeword based on the received codeword and an extrinsic probability measure calculated based on the set of branch transition probabilities, the first and the second probability distribution.

Moreover, the processing means may be adapted to use in either each of or both steps of recalculating the values of the first and second probability distribution a subset of initial values of the first probability distribution or the second probability distribution, respectively, and a subset of the set of branch transition probabilities for recalculating the respective probability distribution, wherein only values are used that fulfill a predetermined reliability criterion.

In a further embodiment of the present invention a decoder comprising means adapted to perform any of the above mentioned decoding methods is provided.

Moreover, another embodiment of the present invention relates to a mobile terminal in a mobile communication system, wherein the mobile terminal may comprise receiving means for receiving at least one codeword, demodulation means for demodulating the at least one received codeword, and a decoder according to one of the embodiments of the present invention.

In another embodiment, the mobile terminal may further comprise encoding means for encoding data in at least one codeword, and transmission means for transmitting the at least one codeword, wherein the at least one transmitted codeword is suitable for decoding according to a decoding methods outlined above.

In a further embodiment of the present invention a base station in a mobile communication system is provided, wherein the base station may comprise receiving means for receiving at least one codeword, demodulation means for demodulating the at least one received codeword, and a decoder according to one of the embodiments of the present invention.

In another embodiment, the base station may further comprise encoding means for encoding data in at least one codeword, and transmission means for transmitting the at least one codeword, wherein the at least one transmitted codeword is suitable for decoding according to a decoding methods outlined above.

Moreover, the according to an even further embodiment provides a mobile communication system comprising at least one base station according to one of the embodiments of the present invention and at least one mobile terminal according to one of the embodiments of the present invention

Brief description of the Figures

In the following the present invention is described in more detail in reference to the attached figures and drawings. Similar or corresponding details in the figures are marked with the same reference numerals.

Fig. 1 shows an exemplary a shift-register encoder layout for systematic encoding,

Fig. 2 shows a state transition diagram of the encoder shown in Fig .1,

Fig. 3 shows a trellis segment description for the encoder shown in Fig. 1,

Fig. 4 shows a trellis segment showing variables for the forward recursion,

Fig. 5 shows a trellis segment showing variables for the backward recursion,

Fig. 6 shows a trellis segment showing variables for the decision,

Fig. 7 shows a flowchart of a decoding process according to one embodiment of the present invention,

Fig. 8 & 9 show flowcharts of a decoding process using the turbo principle according to different embodiments of the present invention,

Fig. 10 shows a transmitter and a receiver unit according to an embodiment of the present invention,

Fig. 11 shows a mobile terminal according to an embodiment of the present invention comprising the transmitter and the receiver shown in Fig. 10,

Fig. 12 shows a base station according to an embodiment of the present invention comprising the transmitter and the receiver shown in Fig. 10, and

Fig. 13 shows an architectural overview of a communication system according to an embodiment of the present invention comprising a mobile terminal shown in Fig. 11 and a base station (Node B) shown in Fig. 12.

Detailed Description

In the following paragraphs the expression "$x \in A \setminus B$" denotes "x is element of set A without set B", which is equivalent to "x is element of set A but not element of set B".

As outlined in the previous sections, mathematical equations may be solved in the initialization, forward recursion, backward recursion, and decision step of the maximum a-posteriori algorithm (see for example Equations 6, 12, 13, 14 and 15).

Generally these equations contain the following terms terms:

- The equation for the initialization contains terms involving y values
- The equation for the forward recursion contains terms involving Γ and determined α values
- The equation for the backward recursion contains terms involving Γ and determined β values

The numerator of Equation 12 for the forward recursion may be interpreted as a sum of values for state transitions which originate in state S_{k-1} terminate in state $S_k = m$. Therefore the following "forward set" can be defined:

$$T_{k,m} = \left\{ S_{k-1} \middle| S_{k-1} \xrightarrow{d_k \in \{0,1\}} S_k = m \right\} \qquad \text{Equation 22}$$

$T_{k,m}$ is the set of states S_{k-1} where transitions from state S_{k-1} to S_k are possible by an information bit d_k.

Therefore

$$\alpha_k(S_k = m) = \log \frac{\sum_{m' \in T_{k,m}} \exp(\alpha_{k-1}(m') + \Gamma_k(y_k, m', m))}{\sum_{m''=1}^{M} \sum_{m' \in T_{k,m''}} \exp(\alpha_{k-1}(m') + \Gamma_k(y_k, m', m''))} \qquad \text{Equation 23}$$

Similarly the numerator of Equation 13 for the backward recursion may be interpreted as a sum of values for state transitions which originate in state S_{k+1} and terminate in state $S_k = m$. Therefore a second "backward set" can be defined:

$$U_{k,m} = \left\{ S_{k+1} \middle| S_k = m \xrightarrow{d_k \in \{0,1\}} S_{k+1} \right\} \qquad \text{Equation 24}$$

$U_{k,m}$ is the set of states S_{k+1} where transitions from state S_k to S_{k+1} are possible by an information bit d_k.

Therefore

$$\beta_k(S_k = m) = \log \frac{\sum_{m'' \in U_{k,m}} \exp(\beta_{k+1}(m'') + \Gamma_{k+1}(y_{k+1}, m, m''))}{\sum_{m'=1}^{M} \sum_{m'' \in U_{k,m'}} \exp(\beta_{k+1}(m'') + \Gamma_{k+1}(y_{k+1}, m', m''))} \qquad \text{Equation 25}$$

According to the present invention, exclusion sets $\Delta_{k,m}$ and $\Omega_{k,m}$ may be additionally defined for the forward and/or backward recursions.

The exclusion set $\Delta_{k,m}$ may indicate those elements in the forward set $T_{k,m}$ that do not fulfill a specific reliability criterion and may therefore not be used in the forward recursion step. Likewise, the exclusion set $\Omega_{k,m}$ may indicate those elements in the backward set $U_{k,m}$ that do not fulfill a specific reliability criterion and may therefore not be used in the backward recursion step.

Employing the exclusion sets $\Delta_{k,m}$ and $\Omega_{k,m}$, the equations may therefore be modified as follows:

New Forward Recursion

$$\alpha_k(S_k=m)=\log\frac{\sum\limits_{m'\in T_{k,m}\setminus\Delta_{k,m}}\exp(\alpha_{k-1}(m')+\Gamma_k(y_k,m',m))}{\sum\limits_{m''=1}^{M}\sum\limits_{m'\in T_{k,m''}\setminus\Delta_{k,m''}}\exp(\alpha_{k-1}(m')+\Gamma_k(y_k,m',m''))}\qquad\text{Equation 26}$$

or alternatively simplified to

$$\alpha_k(S_k=m)=\frac{\max\limits_{m'\in T_{k,m}\setminus\Delta_{k,m}}(\alpha_{k-1}(m')+\Gamma_k(y_k,m',m))}{\max\limits_{m''=1..M}\left[\max\limits_{m'\in T_{k,m''}\setminus\Delta_{k,m''}}(\alpha_{k-1}(m')+\Gamma_k(y_k,m',m''))\right]}\qquad\text{Equation 27}$$

New Backward Recursion

$$\beta_k(S_k=m)=\log\frac{\sum\limits_{m''\in U_{k,m}\setminus\Omega_{k,m}}\exp(\beta_{k+1}(m'')+\Gamma_{k+1}(y_{k+1},m,m''))}{\sum\limits_{m'=1}^{M}\sum\limits_{m''\in U_{k,m'}\setminus\Omega_{k,m'}}\exp(\beta_{k+1}(m'')+\Gamma_{k+1}(y_{k+1},m',m''))}\qquad\text{Equation 28}$$

or alternatively simplified to

$$\beta_k(S_k=m)=\frac{\max\limits_{m''\in U_{k,m}\setminus\Omega_{k,m}}(\beta_{k+1}(m'')+\Gamma_{k+1}(y_{k+1},m,m''))}{\max\limits_{m'=1..M}\left[\max\limits_{m''\in U_{k,m'}\setminus\Omega_{k,m'}}(\beta_{k+1}(m'')+\Gamma_{k+1}(y_{k+1},m',m''))\right]}\qquad\text{Equation 29}$$

If both sets $\Delta_{k,m}$ and $\Omega_{k,m}$ are empty, prior art behavior is replicated. If the exclusion set $\Delta_{k,m}$ contains the same elements as the forward set $T_{k,m}$, then the value of $\alpha_k(S_k=m)$ may not be determined from the recursion formula.

In such a case it may be useful to set the corresponding $\alpha_k(S_k = m) = -\infty$. Likewise $\beta_k(S_k = m) = -\infty$ may be set when the exclusion set $\Omega_{k,m}$ contains the same elements as the backward set $U_{k,m}$.

In case that for a certain value of k an exclusion set is equal to the forward set for all $m = 1\ldots M$, then $\alpha_k(m)$ may be set to $-\ln M$, which means that all states $S_k = 1\ldots M$ are equally likely. The same applies to the backward set.

Generally the exclusion sets may depend for example on the state index m for which an equation is solved, on the information bit index k for which an equation is solved and/or on the iteration number of the decoding procedure (for example in a turbo decoding context).

Definition of exclusion sets $\Delta_{k,m}$ and $\Omega_{k,m}$

As outlined above, the exclusion sets $\Delta_{k,m}$ and $\Omega_{k,m}$ may be defined in order to exclude data from the equations (or decoding process) which are assumed to be wrong, or which are highly likely to be wrong. If such data is included, the produced output is likely to be wrong as well. Therefore the present invention proposes to neglect such values from the equations to overcome their negative impacts on the decoding output.

As mentioned above, the exclusion sets for the new forward recursion step (see Equation 26 or 27) and backward recursion step (see Equation 28 or 29) may be defined such that unreliable messages are excluded from the calculations. In a further embodiment of the present invention the exclusion sets may for example be defined independently from each other, i.e. an element of exclusion set $\Delta_{k,m}$ may not necessarily be element of exclusion set $\Omega_{k,m}$.

Similarly, in another embodiment of the present invention, the exclusion sets $\Delta_{k,m}$ and $\Omega_{k,m}$ may be set independently in decoding iterations. When increasing the number of iterations, the overall reliability of messages passed may be increased for reasonably good transmission conditions. This may be for example applicable to the decoding of turbo codes, where the extrinsic information exchanged between decoding entities usually increases in reliability with an increased number of decoding iterations.

Therefore, when increasing the number of iterations the number of elements of the exclusion sets may be reduced, such that at late stages (in terms of iterations) of decoding the exclusion sets may be empty.

In another embodiment of the present invention the exclusion sets may for example depend both on the number of iterations processed so far, as well as on the maximum number of decoding iterations, which may be a parameter given by the communication system. This may allow a gradual reduction of elements in the exclusion sets depending on the progress of iteration steps.

An exemplary list of possible criteria which may be used isolated or in combination for determining the exclusion sets are channel estimation (signal-to-noise ratio), absolute LLR values, iteration number (in turbo decoding context) and/or a random process.

For example a channel estimation criterion allows the definition of exclusion sets according to the perceived quality of received data. The advantage may be that the channel estimation provides a sort of independent side information known at the decoder to estimate the reliability of received coded information. However, the granularity of a channel estimate may be restricted to a segment which consists of several bits, so this measure alone may not be applicable in all situations to define an exclusion set.

An absolute LLR value criterion may allow reliability estimation with a fine granularity. Due to the definition of the LLR value, large absolute values represent a high confidence. Conversely a small absolute value represents a low confidence. Therefore a ranking of absolute LLR values may be used to determine the smallest values for a given equation to be part of the exclusion set. For example, a LLR value criterion may be used alone or in combination with other criteria to determine the elements in the exclusion sets.

A further possible criterion may be a random process criterion. This criterion may be used either alone or in conjunction with other criteria to determine members of the exclusion set. For example, due to channel estimation it may be assumed that 10% of the received information is unreliable. Then for each piece of information there may be a chance of 10% for being member of an exclusion set.

Next, in reference to Fig. 7, 8 and 9, different embodiments of the present invention will be outlined.

Fig. 7 shows a flowchart of a decoding process according to one embodiment of the present invention. Upon receiving a codeword y_k via the air interface in step 701, the decoder may generate the exclusion sets $\Delta_{k,m}$ and $\Omega_{k,m}$ in step 702.

In order to generate the exclusion sets, several different decision parameters may be used to decide which elements should be excluded from the calculations in the forward recursion and/or backward recursion steps 704, 705. For example, receiving means may provide information on the channel quality for the reception of the codeword or individual bits thereof, or may even provide the exclusion sets $\Delta_{k,m}$ and $\Omega_{k,m}$ to the decoder.

Further, based on the knowledge of the encoder structure and the received codeword y_k the branch transition probabilities $\Gamma(y_k, S_{k-1}, S_k)$ may be initialized in step 703. Also the probability distributions α_k and β_k are initialized in step 704. This may be for example done using the knowledge of the encoder structure used to generate the received codeword y_k.

Having initialized the decoder appropriately, the forward recursion and the backward recursion, as for example defined in Equations 26 to 29, may be performed in steps 705 and 706. In these recursions the exclusion sets $\Delta_{k,m}$ and $\Omega_{k,m}$ are considered, i.e. only a subset of the values in the distributions α_k , β_k and/or $\Gamma(y_k, S_{k-1}, S_k)$ may be used to perform the recursion steps.

Upon having recalculated the new values of α_k and β_k, the codeword may be reconstructed by the decoder. This step may for example include the generation of the extrinsic LLR $L^e(x_k^s)$ and an estimation criterion $L(d_k)$ for deciding upon the individual bits of the decoded codeword $\hat{d}_k$.

In a further embodiment, it may be further possible to reuse the extrinsic LLR $L^e(x_k^s)$ or the estimation criterion $L(d_k)$ as a parameter for the initialization of the branch transition probabilities $\Gamma(y_k, S_{k-1}, S_k)$ of the next decoding procedure for the subsequent codeword. However, this may facilitate the propagation of decoding errors of a previous codeword to the next codeword.

Fig. 8 and 9 show flowcharts of a decoding process using the turbo principle according to further exemplary embodiments of the present invention. In these examples multiple decoder instances are used in the decoder. For example, such a structure may be application for use with turbo encoders/decoders.

The left branch in the Fig. 8 and 9 illustrates the operation of a first decoder instance while the right branch illustrates the operation of the second decoder instance. To better differentiate between the parameters of the two different decoder instances, the 1s and 2s have been added in superscript or subscript.

Essentially, the steps performed by both decoder instances are similar to the respective steps outlined with reference to Fig. 7. In the following description of Fig. 8 and 9, it will be therefore focused on the changes applied to the decoding process.

In Fig. 8, a receiving means receives a codeword y_k in step 801 and may provide same to the first decoder instance. Upon generating or obtaining the exclusions sets $\Delta^1_{k,m}$ and $\Omega^1_{k,m}$ (see step 702), for example using reception quality indicators for the individual bits of a receiving means, the branch transition probabilities $\Gamma^1(y_k, S^1_{k-1}, S^1_k)$ and the values of α^1_k and β^1_k may be initialized (see steps 703 and 704). Next, the forward recursion step 705 and the backward recursion step 706 are executed.

According to this embodiment of the present invention, the first decoder instance may generate extrinsic LLR $L^e_1(x^s_k)$ (or alternatively an estimation criterion $L_1(d_k)$ based thereon) in step 802 instead of reconstructing the codeword $\hat{d}_k$. The generated extrinsic LLR $L^e_1(x^s_k)$ (or the estimation criterion $L_1(d_k)$) may be forwarded to the second decoder instance for use in its decoding process, which will be explained next.

In step 803 the second decoder instance receives the codeword y_k from the receiving means. Next, it may generate the exclusions sets $\Delta^2_{k,m}$ and $\Omega^2_{k,m}$ or may be provided with same. Alternatively, for example, when using the results of the first decoder instance as indicated by the dotted arrow, the exclusions sets $\Delta^2_{k,m}$ and $\Omega^2_{k,m}$ will be generated in step 803. It should be noted that the consideration of the processing results of the first decoder instance is optional in step 803.

Next, the second decoder instance may initialize the branch transition probabilities $\Gamma^2(y_k, S_{k-1}^2, S_k^2)$ in step 804. The extrinsic LLR $L_1^e(x_k^s)$ or the estimation criterion $L_1(d_k)$ may be used as the intrinsic LLR $L_2^i(x_k^s)$ in the initialization in the second decoder instance. Further, the values of α_k^2 and β_k^2 are initialized in a similar manner as described for steps 703 and 704.

Upon initializing $\Gamma^2(y_k, S_{k-1}^2, S_k^2)$, α_k^2 and β_k^2, the forward recursion step 806 and the backward recursion step 807 are executed in a similar manner as described with reference to steps 705 and 706 of Fig. 7.

After having recalculated the probability distributions α_k^2 and β_k^2 the codeword $\hat{d}_k$ may be reconstructed. According to the exemplary embodiment of Fig. 8, the extrinsic LLR $L_2^e(x_k^s)$ may be generated next in step 808 and based in these values the codeword $\hat{d}_k$ may be reconstructed in step 809.

As has become apparent, the second decoder instance may be operated with a delay relative to the first decoder instance, such that the results of the first decoder instance may be used in the decoding procedure of the second decoder instance. It should also be further noted that in an alternative embodiment the first decoder instance may reconstruct a decoded codeword which may be compared to same obtained from the second decoder instance. In this case, the second decoder may or may not be operated delayed to the first decoder instance. This process will be more closely described in reference to Fig. 9 in the following.

Fig. 9 shows a flowchart of a decoding process using the turbo principle according to a further exemplary embodiment of the present invention. The decoding processes in the two decoder instances shown in the left and right branches of Fig. 9 are almost identical. The first decoding iteration in the first decoder instance is similar to the one explained with reference to Fig. 8, i.e. for the first decoding iteration steps 901 and 902 are similar to steps 702 and 703 in Fig. 7 and 9.

Upon initialization and the calculations of the forward recursion an backward recursion (see steps 704, 705, 706), the first decoder instance generates an extrinsic LLR $L_1^e(x_k^s)$ which is provided to the second encoder instance. Further, the first decoding instance construct the decode codeword $\hat{d}_k^1$.

In parallel or with a delay allowing the use of the results of the first decoder instance in step 804 (and optionally step 803), the second decoder instance may perform (steps 803 to 807, 809 and 904) a similar decoding as the first decoder instance or a decoding iteration as described with reference to the second decoder instance in Fig. 8.

At the end of the first decoding iteration, the second decoding instance generates a reconstructed codeword $\hat{d}_k^2$. In step 905, the two generated codeword $\hat{d}_k^1$ and $\hat{d}_k^2$ are compared and if found to be equal the decoding process finishes in step 906.

If however the decision in step 905 comes to a negative result, a further decoding iteration may be performed. In this case the second decoder instance may provide its extrinsic LLR $L_2^e(x_k^s)$ to the first decoder instance (step 904) as indicated by the dotted arrows. Similar to the second decoder instance, the first decoder instance may use this extrinsic information as an intrinsic information, e.g. the intrinsic LLR $L_1^i(x_k^s)$, in the decoding iteration. I.e. the information of the second decoder instance may be used for obtaining a newly initialized set of branch transition probabilities $\Gamma^1(y_k, S_{k-1}^1, S_k^1)$ in step 902 and, optionally, for determining the new exclusion sets $\Delta_{k,m}^1$ and $\Omega_{k,m}^1$ in step 901.

Thus, the decoder may perform several iterations before obtaining similar reconstructed codewords $\hat{d}_k^1$ and $\hat{d}_k^2$, which will end the decoding procedure for received codeword y_k. Further, in case the reconstructed codewords $\hat{d}_k^1$ and $\hat{d}_k^2$ do not match after a predetermined number of iterations, the decoding process may be halted and a decoding error may be signaled to the next processing instance.

Though the exemplary decoding procedure of Fig. 9 has been described with both decoder instances reconstructing a codeword and comparing same, it should be noted that also a procedure as proposed in the embodiment shown in Fig. 8 may be employed together with performing several decoding iterations before reconstructing the codeword.

Next, Fig. 10 will be discussed in more detail. Fig. 10 shows a transmitter and a receiver unit according to an embodiment of the present invention. The transmitter 1001 comprises an encoder 1002 and a transmission means 1003. The transmission means may comprise a modulator for modulating the signals encoded by encoder 1002. As indicated by the dotted arrow, the encoder 1002 is capable of encoding input data into codeword suitable for decoding according to the various embodiments of the decoding process described above. The modulated data may be transmitted by the transmission means 1003 using an antenna as indicated.

The receiver 1004 receiving the encoded signals may comprise a receiving means 1006, which may comprise a demodulator for demodulating the received signals. Upon extracting the y_k values and parameters such as the transmission quality or a reliability criterion for each bit in received codeword y_k in the receiving means 1006, these data may be provided to a decoder 1005, which will consider the data to initialize the decoding process as outlined above.

The decoder 1005 may comprise a processing means 1007, adapted to decode the received data according to the methods described above to produce reconstructed codewords.

Fig. 11 and 12 show a mobile terminal (UE) 1101 and a base station (Node B) 1201 according to different embodiments of the present invention, respectively. The mobile terminal 1101 and the base station may each include a transmitter 1001 and a receiver 1004 as shown in Fig. 10 to perform communications.

Fig. 13 shows an architectural overview of a communication system according to an embodiment of the present invention comprising a mobile terminal 1101 shown in Fig. 11 and a base station (Node B) 1201 shown in Fig. 12.

The overview depicts a UMTS network 1301, which comprises a core network (CN) 1303 and the UMTS terrestrial radio access network (UTRAN) 1302. The mobile terminal 1101 may be connected to the UTRAN 1302 via a wireless link to a Node B 1201. The base stations in the UTRAN 1302 may be further connected to a radio network controller (RNC) 1304. The CN 1303 may comprise a (Gateway) Mobile Switching Center (MSC) for connecting the CN 1303 to a Public Switched Telephone Network (PSTN). The Home Location Register (HLR) and the Visitor Location Register (VLR) may be used to store user related information. Further, the core network may also provide connection to an Internet Protocol-based (IP-based) network through the Serving GPRS Support Node (SGSN) and the Gateway GPRS Support Node (GGSN).

Though exemplary reference to a mobile communication system has been made above, those skilled in the art will notice that the present invention may also be applicable for use in wireless (data) networks, as for example IEEE 802.11, digital video broadcasting, such as DVB, or digital audio broadcasting, as for example DAB or DRM.